I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted by facsimile to the Patent and Trademark Office, facsimile no. (571) 273-8300 at MS Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

Dated: July 17, 2008 Signature: _____
(Kevin J. Canning)

Docket No.: MWS-077RCE3
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
John Micco *et al.*

Application No.: 09/911,819 Confirmation No.: 6291

Filed: July 24, 2001 Art Unit: 2193

For: FUNCTION CALL TRANSLATION Examiner: T. A. Vu

PROPOSED CLAIM AMENDMENTS FOR EXAMINER'S AMENDMENT

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

INTRODUCTORY COMMENTS

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks/Arguments begin on page 103 of this paper.

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) In a computing device, a method comprising:

providing a first function in a first programming language in a library file;

deriving a definition of the first function;

creating description information about the first function using the definition of the first function;

translating the first function in the first programming language into a corresponding function in a second programming language using the definition of the first function;

generating a function library containing the corresponding function in the second programming language;

creating a description file from the library file, the description file including the description information about the first function in the library file, the description information enabling translation of a call to the first function in the first programming language into a call to a corresponding function in the second programming language while avoiding accessing the first function in the first programming language for the translation;

storing the function library and the description file;

translating a first file in the first programming language to a corresponding file in the second programming language, the first file comprising one or more calls to the first function, the translating comprising:

accessing the description information about the first function for each of the one or more calls to the first function; ~~and~~

using the description information to create each ~~call~~ of the one or more calls to the corresponding function in the second programming language while avoiding accessing the first function; and

including the corresponding function in the second programming language from the function library as part of the corresponding file in the second programming language.

2. (Canceled)

3. (Currently Amended) The method of claim 1, wherein creating description information about

the first function comprises: examining the definition of the first function associated with the first programming language; and deriving information about the first function.

4. (Canceled)

5. (Canceled)

6. (Currently Amended) The method of claim 1, in which creating description information about the first function comprises: deriving a number of declared formal inputs to the first function.

7. (Currently Amended) The method of claim 1, in which creating description information about the first function comprises: deriving a number of declared formal outputs to the first function.

8. (Currently Amended) The method of claim 1, in which creating description information about the first function comprises: deriving a scope of the first function.

9. (Currently Amended) The method of claim 1, in which creating description information about the first function comprises: determining whether the first function accepts a variable number of arguments.

10. (Currently Amended) The method of claim 1, in creating description information about the first function comprises: determining whether the first function returns a variable number of results.

11. – 20. (Canceled)

21. (Currently Amended) In a computing device, a method comprising:

providing a library file including functions defined by a first programming language;

creating a function library, the function library including one or more functions defined by a second programming language, each function in the function library being a translated version of a function in the library file;

creating a description file from the library file, the description file including description information about each function in the library file, the description information enabling translation of a call to the function in the first programming language into a call to a corresponding function in the second programming language while avoiding accessing the function in the first programming language for each translation;

storing the function library and the description file; and

translating a program file from the first programming language into the second programming language using the description file and the function library to produce a translated file, the program file in the first programming language comprising one or more calls to a selected function among the functions in the library file, the translating comprising:

retrieving the description information for the selected function from the library file,

examining the description information for the selected function to determine a corresponding function in the second programming language that is in the function library,

~~calling the corresponding function in the second programming language from the function library, and~~

translating each call to the selected function in the first programming language into a call to ~~a~~ the corresponding function in the second programming language in the translated file; and

including the corresponding function in the second programming language from the function library as part of the translated file.

22. (Canceled)

23. (Previously Presented) The method of claim 21, wherein creating a description file comprises:

examining the definition of each function in the library file; and

deriving information about each function.

24. (Canceled)

25. (Canceled)

26. (Currently Amended) The method of claim 21, wherein ~~using the description file~~ translating each call comprises: generating a call through a function evaluation interface for the selected function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the selected function.

27. (Currently Amended) The method of claim 21, wherein ~~using the description file~~ translating each call comprises: generating a call through a function evaluation interface for the selected function if the description information includes a descriptor identifying a return of a variable output argument list from the selected function.

28. (Currently Amended) The method of claim 21, wherein ~~using the description file~~ translating each call comprises: generating a call through a normal interface for the selected function if the description information includes a descriptor identifying a known number of input and output arguments to the selected function.

29. (Currently Amended) A computer program product, tangibly stored on a computer-readable medium, for creating a data file, the product comprising instructions operable to cause a programmable processor to:

obtain a first function in a first programming language in a library file;

derive a definition of the first function;

create description information about the first function using the definition of the first function;

translate the first function in the first programming language into a corresponding function in a second programming language using the definition of the first function;

generate a function library containing the corresponding function in the second programming language;

create a description file from the library file, the description including the description information about the first function in the library file, the description information enabling translation of a call to the first function in the first programming language into a call to a

corresponding function in the second programming language while avoiding accessing the first function in the first programming language for the translation;

store the function library and the file description;

translate a first file in the first programming language to a corresponding file in the second programming language, the first file comprising one or more calls to the first function, said translating causing the processor to:

access the description information about the first function for each of the one or more calls to the first function; ~~and~~

use the description information to create a call to the corresponding function in the second programming language while avoiding accessing the first function; and

include the corresponding function in the second programming language from the function library as part of the corresponding file in the second programming language.

30. (Canceled)

31. (Currently Amended) The product of claim 29, wherein creating description information comprises: examining the definition of the first function associated with the first programming language; and deriving information about the first function.

32. (Original) The product of claim 31, further comprising instructions operable to cause a programmable processor to: use the derived information to create the description information.

33. (Canceled)

34. (Currently Amended) The product of claim 29, in which creating description information comprises: deriving a number of declared formal inputs to the first function.

35. (Currently Amended) The product of claim 29, in which creating description information comprises: deriving a number of declared formal outputs to the first function.

36. (Currently Amended) The product of claim 29, in which creating description information

comprises: deriving a scope of the first function.

37. (Currently Amended) The product of claim 29, in which creating description information comprises: determining whether the first function accepts a variable number of arguments.

38. (Currently Amended) The product of claim 29, in which creating description information comprises: determining whether the first function returns a variable number of results.

39. – 48. (Canceled)

49. (Currently Amended) A computer program product, tangibly stored on a computer-readable medium, for translating function calls, the product comprising instructions operable to cause a programmable processor to:

provide a library file including functions defined by a first programming language;

create a function library , the function library including one or more functions defined by a second programming language, each function in the function library being a translated version of a function in the library file;

create the description file from the library file, the description file including description information about each function in the library file, the description information enabling translation of a call to the function in the first programming language into a call to a corresponding function in the second programming language in a manner that avoids accessing the function in the first programming language for each translation;

store the function library and the description file in a storage; and

translate a program file from the first programming language into the second programming language ~~use~~ using the description file and the function library to produce a translated file, the program file in the first programming language comprising one or more calls to a selected function among the functions in the library file, the translating causing the processor to:

retrieve the description information for the selected function from the library file,

examine the description information for the selected function to determine a corresponding function in the second programming language that is in the function library,

~~call the corresponding function in the second programming language from the function library; and~~

translate each call to the selected function in the first programming language into a call to ~~a~~ the corresponding function in the second programming language in the translated file; and

including the corresponding function in the second programming language from the function library as part of the translated file.

50. (Canceled)

51. (Original) The product of claim 49, wherein creating a description file comprises: examining the definition of each function in the library file; and deriving information about each function.

52. (Canceled)

53. (Canceled)

54. (Currently Amended) The product of claim 49, wherein ~~using the description file comprises~~ the instructions operable to cause a programmable processor to translate the program further comprise: instructions to cause the programmable processor to generate ~~generating~~ a call through a function evaluation interface for the selected function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the selected function.

55. (Currently Amended) The product of claim 49, wherein ~~using the description file comprises~~ the instructions operable to cause the programmable processor to translate the program further comprise: instructions to cause the programmable processor to generate ~~generating~~ a call through a function evaluation interface for the selected function if the description information includes a descriptor identifying a return of a variable output argument list from the selected function.

56. (Currently Amended) The product of claim 49, wherein ~~using the description file comprises~~ the instructions operable to cause the programmable processor to translate the program further comprise: instructions to cause the programmable processor to generate ~~generating~~ a call through

a normal interface for the selected function if the description information includes a descriptor identifying a known number of input and output arguments to the selected function.

REMARKS

In view of the above amendment, applicant believes the pending application is in condition for allowance.

Dated: July 17, 2008

Respectfully submitted,

By_____
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127

(Fax)
Attorney/Agent For Applicant